

# Fondamenti di Informatica

## Meccanica

## Energetica

## Edilizia

## M–Z

**6 crediti**

**Salvatore Filippone**

### **Obiettivi del corso**

Scopo del corso è di introdurre gli studenti alla struttura degli elaboratori elettronici ed alla loro programmazione. In particolare si introdurranno le tecniche elementari di calcolo numerico e di gestione di dati tipiche delle applicazioni di simulazione in ambito ingegneristico.

Nel corso viene usato come linguaggio di riferimento il linguaggio Matlab, uno dei linguaggi di *scripting* più comuni nelle applicazioni ingegneristiche; di esso ne esiste anche una versione *open source* denominata Octave, che è sufficiente ai fini di questo corso introduttivo.

Al termine del corso lo studente sarà in grado di scrivere semplici codici di elaborazione di dati utilizzando gli operatori matriciali propri del linguaggio, strutturando le soluzioni opportunamente, ed effettuando semplici analisi di complessità computazionale sui codici prodotti.

### **Prerequisiti**

Nessuno.

### **Programma del corso**

**Generalità:** Introduzione all'informatica: definizioni di base, algoritmo, rappresentazione dei dati. Architettura di Von Neumann; cenni di storia dell'informatica.

**Elementi del linguaggio Matlab:** Elementi del linguaggio: identificatori, parole chiave. Costanti numeriche. Rappresentazione dei numeri in virgola mobile. Tipi di dato numerico; variabili scalari ed espressioni aritmetiche. Espressioni booleane.

**Sintassi del linguaggio Matlab:** Espressioni booleane. Strutture di controllo: istruzioni condizionali `if, then, else, switch`; cicli: `for` e `while`. Istruzione `break`.

**Tipi di dati derivati:** Vettori e matrici; operatori aritmetici e booleani elemento per elemento; concatenazione di array. Indicizzazione indiretta. Operazioni aritmetiche vettoriali e matriciali: prodotto di matrici. Operatori di riduzione. Cell array e stringhe. Strutture (struct).

**Organizzazione del codice:** Script e funzioni. Spazio dei nomi e visibilità delle variabili. Regole di passaggio dei dati; funzioni con risultati multipli. Input/output: istruzioni di interazione con i file (fopen, fclose, fprintf, fscanf).

**Algoritmi:** Esempi di algoritmi: l'algoritmo di Euclide, divisibilità, numeri primi. Ricorsione: il fattoriale; "wrapper functions" e "tail recursion".

**Grafica:** Semplici operatori di grafica in 2D e 3D in Matlab. Diagrammi lineari e logaritmici.

**Calcolo numerico:** Operatore di divisione per matrici; soluzione di sistemi lineari, problemi ai minimi quadrati, "fit" polinomiale. Soluzione di equazioni non lineari con il metodo di bisezione.

**Complessità computazionale:** Definizione; esempi di complessità logaritmica, polinomiale e combinatoria: la ricerca di dati, la complessità degli operatori matriciali.

**Principi di "problem solving":** Organizzazione degli algoritmi, modularità, operatori ricorrenti di "Folding", visita, costruzione.

**Ordinamento:** Principali metodi: inserzione, quicksort, mergesort, e loro complessità nel caso medio e nel caso peggiore.

## Modalità di esame

L'esame si svolge in due sessioni, una teorica ed una pratica. Durante la sessione teorica viene richiesto di risolvere su carta degli esercizi attinenti al programma del corso; il superamento della prova teorica è condizione vincolante per accedere alla prova pratica. Nella prova pratica, svolta in aula informatica, viene proposto un semplice problema, e si richiede di costruire un programma che lo risolva.

## Materiale didattico

Appunti delle lezioni, esercitazioni, testi di esame precedenti, disponibili al sito web del docente.

## Testo consigliato

S. Attaway: MATLAB: A practical Introduction to Programming and Problem Solving, Butterworth-Heinemann.